CALIFORNIA STATE UNIVERSITY
FULLERTON

ISDS 570
Final Report
MV Portfolio Optimization

Under the Guidance of
Dr. Pawel Kalczynski

GROUP D1:
Riya Murdia
Vivek Sompalli
Nishanth Reddy Vonteddu
Yeshwanth Reddy Thummalapelly
Venkata Sai Bharadwaj Velamakanni

TABLE OF CONTENTS

INTRODUCTION

This project is on optimizing a Mean-Variance (MV) portfolio with stocks selected based on the first letters of last names of all the team members using five years daily data. And then the testing is done on three months of data available in the next year. Part 3 of the course material serves as the basis for this project. The required data is extracted from Yahoo Finance and Quandl. Data from January 1, 2016, to March 26, 2021, is included in the dataset. For cleaning data, steps from ETL process are followed. The cleaned data is further analyzed in R studio. The main goal is to minimize potential risks while maximizing potential returns. Thus, the obtained results are then compared with the SP500TR index to verify the optimization.

BACKGROUND

ETL:
ETL (Extract Transform Load) denotes a data integration procedure involving the acquisition of data from diverse sources, its conversion into a meaningful format, and subsequent loading into a designated location, often a database or data warehouse. This process facilitates the smooth integration and cohesive analysis of data from multiple systems or origins, playing a crucial role in data management and analytics. Our ETL procedure utilizes both R Studio and PostgreSQL.

- **Extract:**
  This marks the initial phase of the ETL process where data is fetched from various origins such as databases, files, and APIs. This stage involves identifying appropriate data sources and specifying the data to retrieve. Following extraction, the data is stored in a staging area or another temporary storage location.

- **Transform:**
  This step follows data extraction. Here, the extracted data undergoes cleaning, structuring, and conversion into a format suitable for analysis. Processes like data cleansing, aggregation, enrichment, and mapping may be employed during this phase. The modified data is temporarily stored for future use.

- **Load:**
  Serving as the final stage of the ETL procedure, loading involves transferring the transformed data into the target system, typically functioning as a data warehouse. Throughout this phase, data integrity and consistency are maintained by aligning the converted data with the data model of the destination system. Once loaded into the target system, the data becomes available for examination, reporting, and decision-making purposes.

This project entails data integration, incorporating the ETL procedure, to gather information from diverse sources, organize it consistently, and load it into a designated system for analysis and reporting purposes.

In R Studio, integration of the SP500 index and the daily adjusted close price for each company in the portfolio occurred for transformation and analysis. The outcomes were amalgamated using the reshape2 package into data frames, structured with rows representing each trading day and columns representing each ticker symbol. Subsequently, to generate a multivariate time series with trade dates as row names, the acquired data was combined with a custom calendar and modified. Daily returns for each ticker were efficiently computed using the CalculateReturns function. These tabular return data were then converted into extensible time series data (XTS) in the final transformation stage, delineating returns for both stocks (assets) and the index (benchmark).

During the final loading phase, the xts-formatted data became accessible for the PortfolioAnalytics package, equipped with diverse analytical functions for data optimization. These functions included computing and visualizing cumulative returns from 2016 to 2020 for the optimal portfolio and individual companies. Additionally, annualized returns for the initial three months of 2021 were calculated to compare the portfolio's performance with that of the SP500. Various stages within the final MV portfolio optimization processes were employed to design and adequately evaluate the return data.

The data from 2016 to 2020 was designated as training data, while the SP500 data was kept separate. The first three months' data served as testing data to evaluate the efficacy of the training data. The PortfolioAnalytics package was utilized for the training data, with configurations aimed at risk reduction and minimizing standard deviation, while adhering to a return constraint targeting a minimum acceptable return (MAR).

OBJECTIVE

The aim of this project is to optimize a Mean-Variance (MV) portfolio using company tickers derived from the last names of group members. The data utilized for this optimization covers a period of five years, from 2016 to 2020, and will be evaluated during the initial three months of 2021. Alongside portfolio optimization, we will compute cumulative and annualized returns for the chosen stock tickers. This examination will offer valuable insights into portfolio performance, aiding in informed decision-making regarding investment strategy. Through consideration of individual stock performance and their interrelations, our objective is to construct a diversified portfolio that maximizes returns while mitigating risk. In summary, this project aims to deliver a thorough analysis of selected stocks and their potential to yield returns for investors.

TECHNOLOGIES AND SOFTWARE

This report discusses three robust tools utilized in the statistical analysis process.

Rstudio and R Language: We employed the R language for statistical computations and graphical representations, along with its integrated development environment, RStudio. R provides a wide range of statistical techniques, graphical capabilities, and import/export functionalities, further expandable through user-created packages. The simplicity of installing and utilizing these packages makes R an attractive choice for analysis tasks. Additionally, R's packaging system facilitates data sharing and preservation, enhancing its utility for research communities.

PostgreSQL: An open-source relational database management system known for its flexibility and adherence to SQL standards. PostgreSQL offers an array of features, including triggers, foreign keys, stored procedures, and transaction support, making it a reliable choice for managing diverse workloads. Its flexibility and scalability enable it to handle various scenarios, from small-scale setups to large data warehouses or web applications with numerous concurrent users.

Microsoft Excel: A widely used spreadsheet tool renowned for its versatility, offering features like pivot tables, graphing tools, and the VBA programming language. Excel enables analysts to organize extensive datasets into workbooks comprising multiple worksheets, facilitating complex calculations and data analysis. In this project, we utilized Excel to extract data from SP500TR, perform data cleaning and organization tasks.

Furthermore, we created a personalized calendar using Excel to enhance the reliability and precision of our analysis by excluding non-trading days from the data. In conclusion, these software applications provide a diverse set of functionalities, assisting researchers and analysts across different domains in conducting data analysis and visualization more proficiently and accurately. By leveraging these tools, individuals can extract valuable insights from intricate datasets and make well-informed decisions grounded in their analyses.

STOCK TICKERS SELECTION

SELECTED STOCK TICKERS		
Name Of The team members	Tickers Symbol	Company
Riya Murdia	MIXT	MiX Telematics Ltd
	MTRN	Materion Corp
	MLCO	Melco Resorts & Entertainment Ltd
Vivek Sompalli	SAIA	Saia Inc
	SKT	Tanger Inc
	SFL	SFL Corporation Ltd
Nishanth Reddy Vonteddu	VIDE	Video Display Corporation
	VBIV	VBI Vaccines Inc
	VEON	VEON Ltd
Venkata Sai Bharadwaj Velamakanni	VALU	Value Line, Inc.
	VOD	Vodafone Group Plc
	VIVO	Meridian Bioscience Inc
Yeshwanth Reddy Thummalapelly	TRN	Trinity Industries Inc
	TAL	TAL Education Group
	TBI	Trueblue Inc

ANALYSIS

Cumulative return chart (2016 - 2020)

The following chart (Figure 1) showcases the cumulative returns of 15 selected stock tickers over a five-year period from 2016 to 2020. The horizontal axis represents the timeline, while the vertical axis measures the cumulative returns in percentage. This visual representation facilitates an analysis of the performance variability, stability, and associated risks of each ticker, providing valuable information for strategic investment planning. It can be deduced from the chart.

Key observations include:

- **High Volatility**: Stocks like **TAL** and **SAIA** display high volatility despite high returns, indicating a higher risk.
- **Low Returns: SKT, TRN and VBIV** exhibit minimal growth, suggesting underperformance compared to others.
- **Stable Investment: VALU** is less volatile and provides moderate returns, ideal for those seeking stability.

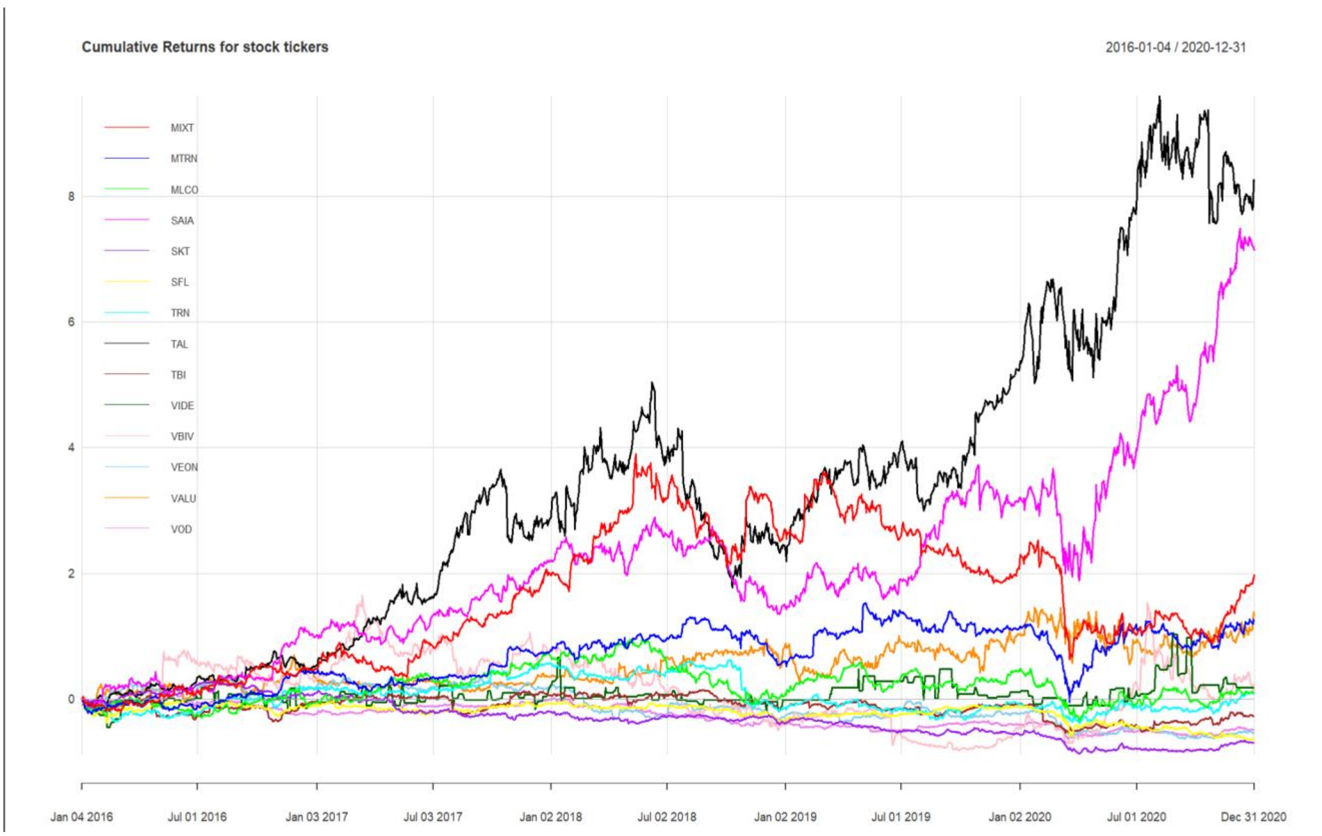

Figure 1: Cumulative Return Chart for 2016 - 2020

Weights of optimized portfolio

Weights in a portfolio indicate how much of the portfolio's total value is invested in each asset or security. This allocation plays a crucial role in determining the portfolio's overall performance and highlights the influence of each asset on the portfolio's results. Essentially, the higher the weight of an asset, the more it can affect the portfolio's returns, either positively or negatively. In our portfolio**, VOD (Vodafone Group Plc)** holds the highest weight, while the **MTRN** and **SKT** carry negative weights, indicating their roles in short selling strategies.

Table 1: Weights of the selected stocks

MXIT	MTRN	MLCO	SAIA	SKT
0.118188	-0.000788	0.012237	0.062727	-0.012032
SFL	TRN	TAL	TBI	VIDE
0.039947	0.02828	0.136985	0.093324	0.044003
VBIV	VEON	VALU	VOD	VIVO
0.016009	0.085921	0.031355	0.237683	0.106162

Cumulative return chart for our optimized portfolio

The following chart displays the cumulative returns for an optimized portfolio (red line) and the SP500 Total Return Index (SP500TR, black line) from January 4, 2021, to March 26, 2021. Key observations include:

- **Performance**: The optimized portfolio significantly outperforms the SP500TR, particularly showing strong gains from late January to early February.
- **Volatility**: The optimized portfolio exhibits greater volatility, with sharp increases and decreases in returns.
- **Correlation and Divergence**: Both indices initially move together, indicating some correlation due to shared economic influences. However, the optimized portfolio frequently diverges, achieving higher peaks, suggesting a more aggressive or effective investment strategy.

In summary, the optimized portfolio provides higher returns with increased risk compared to the more stable but less rewarding SP500TR.

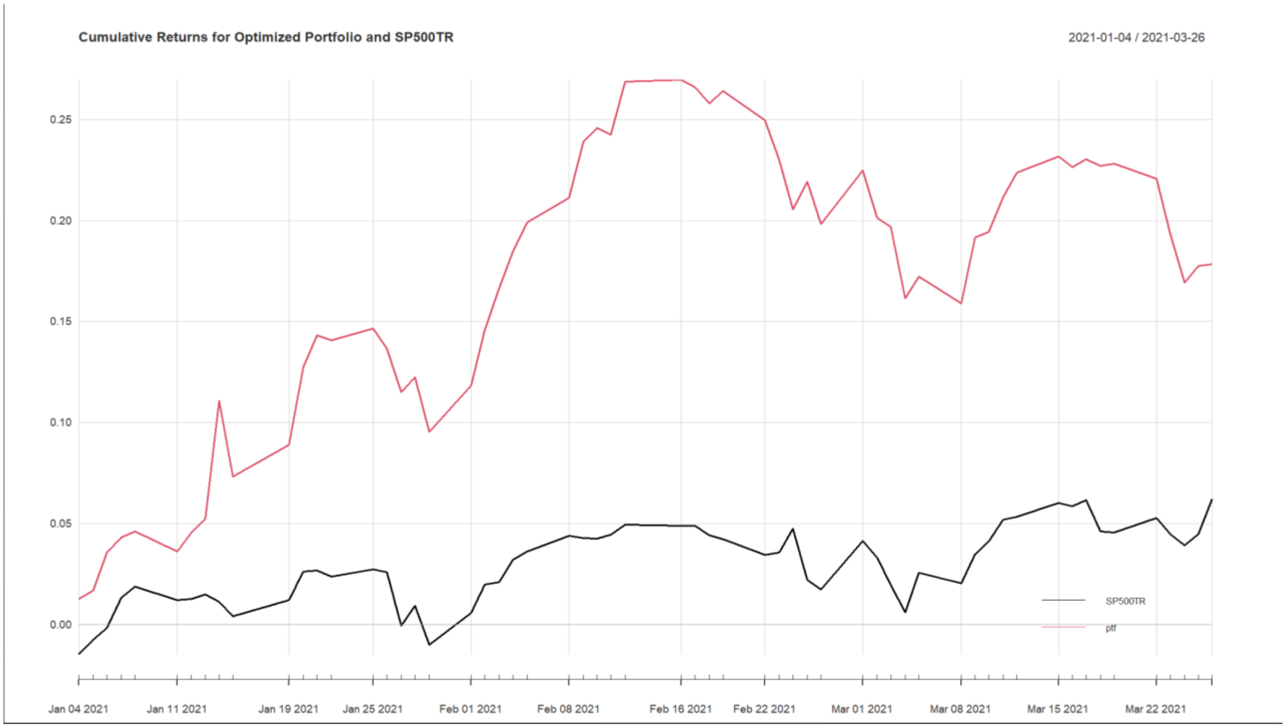

Figure 2: Cumulative Return Chart Comparison For 2021 (Jan to Mar)

Annualized returns for our portfolio and SP500TR index

The table below provides data on the annualized returns, standard deviation, and Sharpe ratio for the SP500 Total Return Index (SP500TR) and our portfolio (ptf). The annualized return for our portfolio is 103.90% as compared to SP500TR which is 29.87%. This indicates that the portfolio has yielded much better returns over the period analyzed. However, this higher return comes with increased volatility, as our portfolio's annualized standard deviation of 0.2624, which is higher than the SP500TR 0.1623. Despite the greater risk, the portfolio's Sharpe ratio of 3.9596 far surpasses the SP500TR 1.8399. The higher Sharpe ratio shows that our portfolio not only delivers higher returns but also does so more efficiently relative to the risk involved. This highlights a successful investment strategy that maximizes returns for each unit of risk taken.

Table 2: Annualized returns for our portfolio and SP500TR index for 2021

	SP500TR	ptf
Annualized Returns	0.2987	1.039
Annualized Standard Deviation	0.1623	0.2624
Annualized Sharp (Rf= 0%)	1.8399	3.9596

CONCLUSION

In this project, we created a portfolio using stock tickers that were chosen based on the initials of group members' last names. The data covered a period from 2016 to 2020, and we evaluated the portfolio's performance over the first three months of 2021. This analysis helped us understand how well our investment choices performed.

Our portfolio achieved a higher annual return compared to the SP 500, indicating that our investment approach was effective. Although our portfolio was more volatile, meaning it had larger fluctuations in value compared to the SP 500, it also had a higher Sharpe ratio. This higher Sharpe ratio means our portfolio not only provided better returns but did so in a way that was more efficient considering the risks involved. Additionally, the cumulative returns show that our portfolio achieved a return of approximately 17.82%, significantly outperforming the SP500 Total Return Index, which had a return of about 6.20%. This indicates that the investment strategy for our portfolio was highly effective, yielding nearly three times the return of the SP 500 during the same period.

Overall, the results are promising and show that our strategy was effective in achieving our goals. Moving forward, it will be important to continue researching to ensure that this strategy can consistently deliver superior results and to determine if it is a good approach for long-term investment.

APPENDIX

```
# Load CSV Files

rm(list=ls(all=T))
setwd("/Users/Project/Desktop/ISDS 570/Group_Project")

# Connect to PostgreSQL

require(RPostgres) # did you install this package?
require(DBI)
conn <- dbConnect(RPostgres::Postgres()
        ,user="stockmarketreader"
        ,password="read123"
        ,host="localhost"
        ,port=5432
        ,dbname="stockmarker_gp"
)

#custom calendar

qry<-"SELECT * FROM custom_calendar WHERE date BETWEEN '2015-12-31' AND '2021-12-31'
ORDER by date"
ccal<-dbGetQuery(conn,qry)

#eod prices and indices

qry1="SELECT symbol,eod_indices.date,adj_close FROM eod_indices INNER JOIN custom_calendar
ON eod_indices.date = custom_calendar.date WHERE eod_indices.date BETWEEN '2015-12-31' AND
'2021-12-31'"
qry2="SELECT ticker,eod_quotes.date,adj_close FROM eod_quotes INNER JOIN custom_calendar ON
eod_quotes.date = custom_calendar.date WHERE eod_quotes.date BETWEEN '2015-12-31' AND '2021-
12-31'"
eod<-dbGetQuery(conn,paste(qry1,'UNION',qry2))
dbDisconnect(conn)
rm(conn)

#Explore

head(ccal)
tail(ccal)
nrow(ccal)
head(eod)
tail(eod)
nrow(eod)
head(eod[which(eod$symbol=='SP500TR'),])
eod_row<-data.frame(symbol='SP500TR',date=as.Date('2015-12-31'),adj_close=3821.60)
eod<-rbind(eod,eod_row)
tail(eod)
```

```
# Use Calendar

tdays<-ccal[which(ccal$trading==1),,drop=F]
head(tdays)
nrow(tdays)-1 #trading days between 2015 and 2020

# Completeness
# Percentage of completeness

pct<-table(eod$symbol)/(nrow(tdays)-1)
selected_symbols_daily<-names(pct)[which(pct>=0.99)]
eod_complete<-eod[which(eod$symbol %in% selected_symbols_daily),,drop=F]

#check

head(eod_complete)
tail(eod_complete)
nrow(eod_complete)

# Transform (Pivot)

require(reshape2) #did you install this package?
eod_pvt<-dcast(eod_complete, date ~ symbol,value.var='adj_close',fun.aggregate = mean, fill=NULL)

#check

eod_pvt[1:10,1:5] #first 10 rows and first 5 columns
ncol(eod_pvt) # column count
nrow(eod_pvt)

#Merge with Calendar

eod_pvt_complete<-merge.data.frame(x=tdays[,'date',drop=F],y=eod_pvt,by='date',all.x=T)

#check

eod_pvt_complete[1:10,1:5] #first 10 rows and first 5 columns
ncol(eod_pvt_complete)
nrow(eod_pvt_complete)

#use dates as row names and remove the date column

rownames(eod_pvt_complete)<-eod_pvt_complete$date
eod_pvt_complete$date<-NULL #remove the "date" column


#re-check

eod_pvt_complete[1:10,1:5] #first 10 rows and first 5 columns
ncol(eod_pvt_complete)
nrow(eod_pvt_complete)
```

```
# Missing Data Imputation
# We can replace a few missing (NA or NaN) data items with previous data
# Let's say no more than 3 in a row.

require(zoo)
eod_pvt_complete<-na.locf(eod_pvt_complete,na.rm=F,fromLast=F,maxgap=3)
#re-check
eod_pvt_complete[1:10,1:5] #first 10 rows and first 5 columns
ncol(eod_pvt_complete)
nrow(eod_pvt_complete)

# Calculating Returns

require(PerformanceAnalytics)
eod_ret<-CalculateReturns(eod_pvt_complete)

#check

eod_ret[1:10,1:3] #first 10 rows and first 3 columns
ncol(eod_ret)
nrow(eod_ret)

#remove the first row

eod_ret<-tail(eod_ret,-1) #use tail with a negative value

#check

eod_ret[1:10,1:3] #first 10 rows and first 3 columns
ncol(eod_ret)
nrow(eod_ret)

# Check for extreme returns
# There is colSums, colMeans but no colMax so we need to create it

colMax <- function(data) sapply(data, max, na.rm = TRUE)

# Apply it

max_daily_ret<-colMax(eod_ret)
max_daily_ret[1:10] #first 10 max returns

# And proceed just like we did with percentage (completeness)

selected_symbols_daily<-names(max_daily_ret)[which(max_daily_ret<=1.00)]
length(selected_symbols_daily)
```

```
#subset eod_ret

eod_ret<-eod_ret[,which(colnames(eod_ret) %in% selected_symbols_daily),drop=F]

#check

eod_ret[1:10,1:3] #first 10 rows and first 3 columns
ncol(eod_ret)
nrow(eod_ret)


# Tabular Return Data Analytics
# We will select 'SP500TR' and 15 RANDOM TICKERS
# We need to convert data frames to xts (extensible time series)

Ra<as.xts(eod_ret[,c('MIXT','MTRN','MLCO','SAIA','SKT','SFL','TRN','TAL','TBI','VIDE','VBIV','VEON','VALU','VOD','VIVO'),drop=F])
Rb<-as.xts(eod_ret[,'SP500TR',drop=F]) #benchmark

# MV Portfolio Optimization
# withhold the last 58 trading days

Ra_training<-head(Ra,-58)
Rb_training<-head(Rb,-58)

# use the last 58 trading days for testing

Ra_testing<-tail(Ra,58)
Rb_testing<-tail(Rb,58)

# Assuming 'Ra_training' is your dataset and 'colors' is a vector of 14 colors
# Define 14 different colors

colors <- c("red", "blue", "green", "magenta", "purple", "yellow", "cyan", "black", "brown", "darkgreen",
"pink", "skyblue", "darkorange","gray75", "violet")

#1 Cumulative return chart for Project Range #1 (2016-2020) for stock tickers selected by our team.

acc_Ra_training<-Return.cumulative(Ra_training);acc_Ra_training
chart.CumReturns(Ra_training, colors = colors, legend.loc = 'topleft', main = 'Cumulative Returns for stock tickers')
```

```
#optimize the MV (Markowitz 1950s) portfolio weights based on training

table.AnnualizedReturns(Rb_training)
mar<-mean(Rb_training) #we need daily minimum acceptable return
require(PortfolioAnalytics)
require(ROI) # make sure to install it
require(ROI.plugin.quadprog)  # make sure to install it
pspec<-portfolio.spec(assets=colnames(Ra_training))
pspec<-add.objective(portfolio=pspec,type="risk",name='StdDev')
pspec<-add.constraint(portfolio=pspec,type="full_investment")pspec<
add.constraint(portfolio=pspec,type="return",return_target=mar)


#optimize portfolio

opt_p<-optimize.portfolio(R=Ra_training,portfolio=pspec,optimize_method = 'ROI')

#2 Weights of your optimized portfolio and the sum of these weights
#extract weights (negative weights means shorting)

opt_w<-opt_p$weights; opt_w
sum_opt_w<-sum(opt_w); sum_opt_w

#apply weights to test returns

Rp<-Rb_testing # easier to apply the existing structure

#define new column that is the dot product of the two vectors

Rp$ptf<-Ra_testing %*% opt_w

#3 Cumulative return chart for your optimized portfolio and SP500TR index

Return.cumulative(Rp)
chart.CumReturns(Rp, legend.loc = 'bottomright', main = 'Cumulative Returns for Optimized Portfolio
and SP500TR')

#4 Annualized returns for your portfolio and SP500TR index

table.AnnualizedReturns(Rp)
```